

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-039937

(43)Date of publication of application : 08.02.2000

(51)Int.Cl. G06F 1/32  
G06F 1/00  
G06F 1/04  
G06F 15/78

(21)Application number : 10-206637

(71)Applicant : TOSHIBA CORP

(22)Date of filing : 22.07.1998

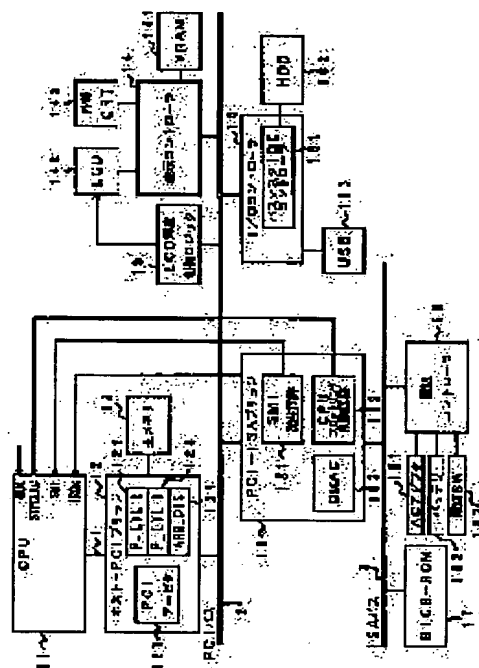
(72)Inventor : MAEDA MAYUMI  
NAKAMURA KOJI

## (54) COMPUTER SYSTEM AND ITS POWER-SAVING CONTROL METHOD

### (57)Abstract:

**PROBLEM TO BE SOLVED:** To prolong a battery-driven operation time by saving the electric power sufficiently while minimizing the degradation of system performance.

**SOLUTION:** When a wake-up event is generated while a CPU 11 is held in a sleep state, the operation speed of the CPU 11 is not put immediately back to the operation speed before the CPU enters the sleep state, but increased gradually by steps from a low level to a specific high level at specific time intervals under the throttling control of a CPU throttling control circuit 152. Consequently, the power consumption of the CPU is reducible as compared with a case wherein the operation speed of the CPU 11 is put directly back to the operation speed before the CPU enters the sleep state and the battery-driven operation time can be prolonged.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

THIS PAGE BLANK (USPTO)

特開2000-39937

(P 2 0 0 0 - 3 9 9 3 7 A)

(43)公開日 平成12年2月8日(2000.2.8)

(51) Int. Cl. 7	識別記号	F I	ターコード	(参考)
G06F 1/32		G06F 1/00	332 Z	5B011
1/00	370		370 D	5B062
1/04	301	1/04	301 C	5B079
15/78	510	15/78	510 P	

審査請求 未請求 請求項の数11 O.L. (全14頁)

(21)出願番号	特願平10-206637	(71)出願人	000003078 株式会社東芝 神奈川県川崎市幸区堀川町72番地
(22)出願日	平成10年7月22日(1998.7.22)	(72)発明者	前田 真弓 東京都青梅市末広町2丁目9番地 株式会社東芝青梅工場内
		(72)発明者	中村 浩二 東京都青梅市末広町2丁目9番地 株式会社東芝青梅工場内
		(74)代理人	100058479 弁理士 鈴江 武彦 (外6名)

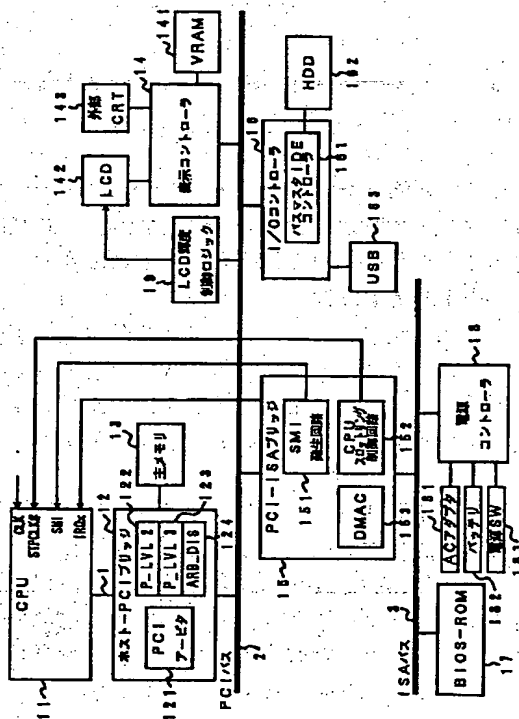
最終頁に続く

(54) 【発明の名称】 コンピュータシステムおよびそのパワーセーブ制御方法

(57) 【要約】

【課題】システム性能の低下を最小限に抑えつつ十分な  
パワーセーブを実現できるようにし、バッテリーによる動  
作時間を延ばす。

【解決手段】CPU11がスリープ状態に維持されている状態でウェイクアップイベントが発生したとき、CPU11の動作速度はすぐにスリープ状態移行前の動作速度に復帰されるのではなく、CPUスロットリング制御回路152によるスロットリング制御により、所定の時間間隔で段階的に低レベルから所定の高レベルにまで徐々に上昇される。これにより、CPU11の動作速度をすぐにスリープ状態移行前の動作速度に復帰させる場合に比し、CPUの電力消費を少なくすることが可能となり、バッテリー動作時間を延ばすことができる。



## 【特許請求の範囲】

【請求項 1】 バッテリ駆動可能なコンピュータシステムにおいて、

前記コンピュータシステムのアイドル時に、前記コンピュータシステムを、動作状態からスリープ状態に移行させるスリープ手段と、

所定のウェイクアップイベントの発生に応答して前記コンピュータシステムが前記スリープ状態から前記動作状態に復帰したとき、前記コンピュータシステムの処理速度を低レベルから所定の高レベルにまで段階的に上昇させる処理速度制御手段とを具備することを特徴とするコンピュータシステム。

【請求項 2】 前記スリープ手段は、前記コンピュータシステムの CPU を、命令実行可能な動作状態から、命令実行が停止される低消費電力のスリープ状態に移行させる CPU スリープ手段を含み、

前記処理速度制御手段は、前記 CPU が前記スリープ状態から前記動作状態に復帰したとき、前記 CPU の動作速度を低レベルから所定の高レベルにまで所定の時間間隔で段階的に上昇させる CPU 速度制御手段を含み、

前記 CPU がスリープ状態に維持されているアイドル期間と前記 CPU が動作状態に維持されているウェイク期間との比率に基づいて、前記コンピュータシステムの稼働率が所定のしきい値以下であるか否かを検出する稼働率検出手段をさらに具備し、

前記 CPU 速度制御手段による前記段階的な動作速度制御処理は、前記コンピュータシステムの稼働率が所定のしきい値以下であることが検出されたときにその実行が許可されることを特徴とする請求項 1 記載のコンピュータシステム。

【請求項 3】 前記スリープ手段は、前記コンピュータシステムの CPU を、命令実行可能な動作状態から、命令実行が停止される低消費電力のスリープ状態に移行させる CPU スリープ手段を含み、

前記処理速度制御手段は、前記 CPU が前記スリープ状態から前記動作状態に復帰したとき、前記 CPU の動作速度を低レベルから所定の高レベルにまで所定の時間間隔で段階的に上昇させる CPU 速度制御手段を含み、

前記 CPU 速度制御手段は、所定のレジスタに設定されたデューティ制御情報に基づいて、前記 CPU のクロックを制御するためのストップクロック信号のデューティ比を複数段階に可変設定して前記 CPU に供給する CPU スロットリング制御手段と、

前記デューティ制御情報の更新要求を示す割り込み信号を所定の時間間隔で前記 CPU に発生する手段と、

前記 CPU の動作速度が低レベルから所定の高レベルにまで段階的に上昇されるように、前記割り込み信号の発生の度に前記デューティ制御情報を更新する手段とを具備することを特徴とする請求項 1 記載のコンピュータシ

ステム。

【請求項 4】 前記スリープ手段は、前記コンピュータシステムの CPU を、命令実行可能な動作状態から、命令実行が停止される低消費電力のスリープ状態に移行させる CPU スリープ手段を含み、

前記処理速度制御手段は、前記 CPU が前記スリープ状態から前記動作状態に復帰したとき、前記 CPU の動作速度を低レベルから所定の高レベルにまで所定の時間間隔で段階的に上昇させる CPU 速度制御手段を含み、

10 前記 CPU は、第 1 のスリープ状態と、この第 1 のスリープ状態よりも低消費電力の第 2 のスリープ状態とを有し、

前記 CPU スリープ手段は、

前記コンピュータシステムの状態が、前記第 2 のスリープ状態への移行が許可される所定の条件を満足しているか否かを判別する手段と、

前記条件が満足されているとき前記 CPU を前記第 2 のスリープ状態に移行させ、前記条件が満足されないとき前記 CPU を前記第 1 のスリープ状態に移行させる手段とを含むことを特徴とする請求項 1 記載のコンピュータシステム。

【請求項 5】 前記第 1 のスリープ状態は、前記 CPU がキャッシュの整合性を維持することが可能な状態であり、前記第 2 のスリープ状態は、前記 CPU がキャッシュの整合性を維持することが不可能な状態であり、前記 CPU スリープ手段は、

前記コンピュータシステムにおいて前記 CPU 以外の他のデバイスがバスマスタ動作しているか否かに基づいて、前記第 2 のスリープ状態への移行が許可される所定の条件を満足しているか否かを判定することを特徴とする請求項 4 記載のコンピュータシステム。

【請求項 6】 バッテリ駆動可能なコンピュータシステムにおいて、

命令実行が停止される第 1 のスリープ状態と、命令実行が停止され、且つ前記第 1 のスリープ状態よりも低消費電力の第 2 のスリープ状態とを有する CPU と、

前記コンピュータシステムのアイドル時に、前記 CPU を、命令実行可能な動作状態から、前記第 1 および第 2 のスリープ状態のいずれか一方のスリープ状態に移行させる CPU スリープ手段とを具備し、

この CPU スリープ手段は、

前記コンピュータシステムの状態が、前記第 2 のスリープ状態への移行が許可される所定の条件を満足しているか否かを判別する手段と、

前記条件が満足されているとき、前記 CPU を前記第 2 のスリープ状態に移行させ、前記条件が満足されないとき前記 CPU を前記第 1 のスリープ状態に移行させる手段とを含むことを特徴とするコンピュータシステム。

【請求項 7】 前記第 1 のスリープ状態は、前記 CPU がキャッシュの整合性を維持することが可能な状態であ

り、前記第2のスリープ状態は、前記CPUがキャッシュの整合性を維持することが不可能な状態であり、前記CPUスリープ手段は、

前記コンピュータシステムにおいて前記CPU以外の他のデバイスがバスマスタ動作しているか否かに基づいて、前記第2のスリープ状態への移行が許可される所定の条件を満足しているか否かを判定することを特徴とする請求項6記載のコンピュータシステム。

【請求項8】 コンピュータシステムのパワーセーブ制御方法であって、

前記コンピュータシステムのアイドル時に、前記コンピュータシステムを動作状態からスリープ状態に移行させるステップと、

所定のウェイクアップイベントの発生にตอบสนองして前記コンピュータシステムが前記スリープ状態から前記動作状態に復帰したとき、前記コンピュータシステムの処理速度を、低レベルから所定の高レベルにまで段階的に上昇させるステップとを具備することを特徴とするパワーセーブ制御方法。

【請求項9】 前記コンピュータシステムを動作状態からスリープ状態に移行させるステップは、前記コンピュータシステムのCPUを、命令実行可能な動作状態から、命令実行が停止される低消費電力のスリープ状態に移行させるステップを含み、

前記処理速度を段階的に上昇させるステップは、前記CPUが前記スリープ状態から前記動作状態に復帰したとき、前記CPUの動作速度を低レベルから所定の高レベルにまで所定の時間間隔で段階的に上昇させるステップを含み、

前記CPUがスリープ状態に維持されているアイドル期間と前記CPUが動作状態に維持されているウェイク期間との比率に基づいて、前記コンピュータシステムの稼働率が所定のしきい値以下であるか否かを検出するステップをさらに具備し、

前記コンピュータシステムの稼働率が所定のしきい値以下であることが検出されたとき、前記CPUの動作速度を、低レベルから所定の高レベルにまで所定の時間間隔で段階的に上昇させることを特徴とする請求項8記載のパワーセーブ制御方法。

【請求項10】 命令実行が停止される第1のスリープ状態と、命令実行が停止され、且つ前記第1のスリープ状態よりも低消費電力の第2のスリープ状態とを有するCPUを備えたコンピュータシステムのパワーセーブ制御方法であって、

前記コンピュータシステムのアイドル時に、前記コンピュータシステムの状態が、前記第2のスリープ状態への移行が許可される所定の条件を満足しているか否かを判別し、

前記条件が満足されているとき、前記CPUを、命令実行可能な動作状態から前記第2のスリープ状態に移行さ

せ、前記条件が満足されないとき前記CPUを前記動作状態から前記第1のスリープ状態に移行させることを特徴とするパワーセーブ制御方法。

【請求項11】 前記第1のスリープ状態は、前記CPUがキャッシュの整合性を維持することが可能な状態であり、前記第2のスリープ状態は、前記CPUがキャッシュの整合性を維持することが不可能な状態であり、前記コンピュータシステムにおいて前記CPU以外の他のデバイスがバスマスタ動作しているか否かに基づいて、

10 前記第2のスリープ状態への移行が許可される所定の条件を満足しているか否かを判定することを特徴とする請求項10記載のパワーセーブ制御方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明はバッテリー駆動可能なパーソナルコンピュータなどのコンピュータシステムに関し、特にそのパワーセーブのための機能を持つコンピュータシステムに関する。

【0002】

20 【従来の技術】近年、携行が容易でバッテリーにより動作可能なラップトップタイプまたはノートブックタイプのパーソナルコンピュータ（PC）が種々開発されている。この種のPCに於いては、CPUの高性能化が進められており、これによってユーザは快適な使用環境を容易に手にすることが可能になってきている。

【0003】ところが、CPUの高性能化に伴い、CPUの消費電力も大きくなり、これによりPC全体の電力消費量の増大およびバッテリー動作時間の低下などの問題が生じている。

30 【0004】そこで、最近では、各種のパワーマネージメント技術が開発され始めている。PCで用いられている代表的なパワーマネージメント技術には、LCDの輝度制御、HDDの自動モータオフ、およびCPUスリープ制御などがある。

【0005】しかし、このような従来のパワーマネージメントは、パワーセーブ機能（LCDの輝度制御、HDDの自動モータオフ、CPUスリープ制御など）を使用するか、使用しないかをユーザ設定などによって予め固定的に決めて行うものであった。

40 【0006】これは、パワーセーブとシステム性能とのバランスを取る上での悩みの種であり、システムとしての性能の障害となるものであった。つまり、パワーセーブ機能を高めるためには、代わりにシステムの性能を常に（バッテリー残量やシステムの稼働状況等に関係なく）低く抑えることを容認しなくてはならず、逆に、システム性能を高く保つと、今度は、パワーセーブ機能がある程度諦めなくてはならなかった。

50 【0007】また、従来の典型的なCPUスリープ制御では、アイドル時にCPUステートを低消費電力のスリープ状態に移行し、割り込みなどのウェイクイベントの

発生時にCPUステートを基の状態に復帰させるという制御が行われる。この場合、ウェイクイベントが一旦発生すると、CPUはすぐにスリープ状態移行前の性能に復帰される。このため、たとえばユーザによってキーボードやマウス操作が一度行われただけで、CPUは、最大性能あるいはそれに近い予め決められた性能に復帰されてしまい、そしてその性能は再びアイドルが検出されるまで維持される。したがって、特にシステム稼働率が比較的低い場合においては、多くの無駄な電力が消費されることになる。

【0008】また、スリープ状態として使用されるCPUステートについても常に固定的に決められていた。すなわち、ACPI (Advanced Configuration and Power Management Interface Specification) 仕様では、CPUパワーステートとして、通常の動作状態として使用されるC0ステートの他に、スリープ状態として使用可能なC1～C3の3つのステートが定義されている。これらパワーステートC1～C3は、C0ステートへの復帰までのレイテンシおよびスリープの深さが異なっており、C1、C2、C3の順で消費電力は小さくなり、C0ステートへの復帰までのレイテンシはC1、C2、C3の順で大きくなる。

【0009】しかし、従来では、ACPI-OSのない環境下では、スリープ状態としてC1～C3の中のどのステートを使用するかは、システムによって予め固定的に規定されており、システムの動作状態などに応じて使用するパワーステートを動的に切り替えるという制御は行われてない。このため、適切なCPUパワーステートを選択することができず、より深いスリープ状態に移行可能な場合であっても浅いスリープしか利用できないなどの問題があった。

#### 【0010】

【発明が解決しようとする課題】上述のように、従来では、パワーセーブを行うためにはシステムの性能を固定的に低下させる必要があり、その為、システム性能とパワーセーブとをバランス良く両立させることは困難であった。特に、CPUスリープ制御においては、ウェイクイベントの発生にตอบสนองしてCPU性能がすぐにスリープ状態移行前の高性能の状態に復帰されてしまい、無駄な電力が消費されることがあった。さらに、アイドル時に移行させるCPUスリープ状態の深さも常に固定されており、十分なパワーセーブを実現することは困難であった。

【0011】本発明はこのような点に鑑みてなされたものであり、システム性能の低下を最小限に抑えつつ十分なパワーセーブを実現できるようにし、バッテリーによる動作時間を大幅に延長することが可能なコンピュータシステムおよびそのパワーセーブ制御方法を提供することを目的とする。

#### 【0012】

【課題を解決するための手段】上述の課題を解決するため、本発明は、バッテリー駆動可能なコンピュータシステムにおいて、前記コンピュータシステムのアイドル時に、前記コンピュータシステムを、動作状態からスリープ状態に移行させるスリープ手段と、所定のウェイクアップイベントの発生にตอบสนองして前記コンピュータシステムが前記スリープ状態から前記動作状態に復帰したとき、前記コンピュータシステムの処理速度を低レベルから所定の高レベルにまで段階的に上昇させる処理速度制御手段とを具備することを特徴とする。

【0013】前記スリープ手段としては、前記コンピュータシステムのCPUを、命令実行可能な動作状態から、命令実行が停止される低消費電力のスリープ状態に移行させるCPUスリープ手段を使用することが好ましい。また、前記処理速度制御手段としては、前記CPUが前記スリープ状態から前記動作状態に復帰したとき、前記CPUの動作速度を低レベルから所定の高レベルにまで所定の時間間隔で段階的に上昇させるCPU速度制御手段を使用することが好ましい。

【0014】このコンピュータシステムにおいては、例えばCPUがスリープ状態に維持されている状態でウェイクアップイベントが発生したとき、CPUの動作速度はすぐにスリープ状態移行前の動作速度に復帰されるのではなく、段階的に低レベルから所定の高レベルにまで上昇される。これにより、システムの処理速度は、段階的に上昇され、処理速度をすぐにスリープ状態移行前の処理速度に復帰させる場合に比し、電力消費を少なくすることが可能となる。また、動作状態への復帰タイミング自体の遅れはない。したがって、スリープ状態の期間中にたとえばユーザによってキーボードやマウスの操作が行われた場合でも、即座にそれに対応する処理を実行することができる。また、ある一定期間後にはシステム処理性能は基の性能に復帰される。このため、過大なCPU負荷を伴う処理などが突然要求されない限りは、ユーザによる体感速度が低下されるといった不具合は一切生じない。よって、システム性能の低下を最小限に抑えつつ十分なパワーセーブを実現することができ、バッテリーによる動作時間を大幅に延長することが可能となる。

【0015】また、本発明は、前記CPUがスリープ状態に維持されているアイドル期間と前記CPUが動作状態に維持されているウェイク期間との比率に基づいて前記コンピュータシステムの稼働率が所定のしきい値以下であるか否かを検出する稼働率検出手段をさらに具備し、前記CPU速度制御手段による前記段階的な動作速度制御処理は、前記コンピュータシステムの稼働率が所定のしきい値以下であることが検出されたときにその実行が許可されることを特徴とする。

【0016】これにより、システム稼働率が比較的低い場合にのみ前述の段階的な動作速度制御処理を実行でき

るようになり、システム性能とパワーセーブとのバランスをより良好に保つことが可能となる。

【0017】また、前記CPU速度制御手段は、所定のレジスタに設定されたデューティ制御情報に基づいて、前記CPUのクロックを制御するためのストップクロック信号のデューティ比を複数段階に可変設定して前記CPUに供給するCPUスロットリング制御手段と、前記デューティ制御情報の更新要求を示す割り込み信号を所定の時間間隔で前記CPUに発生する手段と、前記CPUの動作速度が低レベルから所定の高レベルにまで段階的に上昇されるように、前記割り込み信号の発生の度に前記デューティ制御情報を更新する手段とから構成することが好ましい。

【0018】このようにCPUへのハードウェア割り込み信号を利用して、動作状態におけるCPUの性能を段階的に上昇させることにより、実行中のOSやアプリケーションに影響を与えることなく、CPU性能を効率よく段階的に上昇させることが可能となる。

【0019】また、CPUが、第1のスリープ状態と、この第1のスリープ状態よりも低消費電力の第2のスリープ状態とを有する場合には、前記CPUスリープ手段は、前記コンピュータシステムの状態が、前記第2のスリープ状態への移行が許可される所定の条件を満足しているか否かを判別する手段と、前記条件が満足されているとき前記CPUを前記第2のスリープ状態に移行させ、前記条件が満足されないとき前記CPUを前記第1のスリープ状態に移行させる手段とを含むことを特徴とする。

【0020】このようにコンピュータシステムの状態に応じて移行すべきスリープ状態を選択することにより、より適切なスリープ状態を選択できるようになる。さらに、前記第1のスリープ状態は、前記CPUはキャッシュの整合性を維持することが可能な状態であり、前記第2のスリープ状態は、前記CPUはキャッシュの整合性を維持することが不可能な状態である場合においては、前記コンピュータシステムにおいて前記CPU以外の他のデバイスがバスマスタ動作しているか否かを、前記第2のスリープ状態への移行が許可される所定の条件として利用することができる。このように、バスマスタ動作しているデバイスが存在しないことを条件に第2のスリープ状態に移行することにより、バスマスタ動作しているデバイスによる主メモリの書き換えによって主メモリとCPUキャッシュの不整合が生じるといった問題の発生を防止することが出来る。

【0021】

【発明の実施の形態】以下、図面を参照して本発明の実施形態を説明する。図1には、本発明の一実施例に係わるコンピュータシステムの構成が示されている。このコンピュータシステムはバッテリー駆動可能なノートブックタイプのパーソナルコンピュータ(PC)であり、AC

アダプタ181を介して外部電源が供給されている場合にはその外部電源によって動作すると共に、バッテリー182の充電が行われる。一方、モバイル環境で使用される場合など、PC本体にACアダプタ181が接続されていない状態においては、PCはバッテリー182からの電源によって動作する。

【0022】このPC本体には、図示のように、プロセッサバス1、PCIバス2、ISAバス3、CPU11、ホスト-PCIブリッジ12、主メモリ13、表示コントローラ14、PCI-ISAブリッジ15、I/Oコントローラ16、BIOS-ROM17、電源コントローラ18、およびLCD輝度制御ロジック19などが設けられている。

【0023】CPU11としては、例えば、米インテル社により製造販売されているマイクロプロセッサ“Pentium”などが使用される。CPU11はPLL回路を内蔵しており、このPLL回路は外部クロックCLKに基づいてその外部クロックCLKと同一またはそれよりも高速の内部クロックCLK2を生成する。このCPU11は、ストップクロック信号STPCLK#によってそのクロック状態が制御されるように構成されており、ストップクロック信号STPCLK#によって、3つのクロックステート、つまり、ノーマルステート(Normal State)、ストップグラントステート(Stop Grant State)、およびストップクロックステート(STOP Clock State)を有している。

【0024】ノーマルステートはCPU11の通常の動作ステートであり、命令はこのノーマルステートにおいて実行される。このノーマルステートは電力消費の最も多いステートであり、その消費電流は~700mA程度である。

【0025】最も電力消費の少ないのはストップクロックステートであり、その消費電流は~30μA程度である。このストップクロックステートにおいては、命令の実行が停止されるだけでなく、外部クロックCLKおよび内部クロックCLK2も停止されている。

【0026】ストップグラントステートは、ノーマルステートとストップクロックステートの中間の動作ステートであり、その消費電流は20~55mA程度と比較的に少ない。ストップグラントステートにおいては、命令は実行されない。また、外部クロックCLKおよび内部クロックCLK2は共にランニング状態であるが、CPU内部ロジック(CPUコア)への内部クロックCLK2の供給は禁止される。このストップグラントステートは外部クロックCLKの停止可能なステートであり、このストップグラントステートにおいて外部クロックCLKを停止すると、CPU11はストップグラントステートからストップクロックステートに移行する。

【0027】ノーマルステートとストップグラントステ

ート間の遷移は、ストップクロック (STPCLK#) 信号によって高速に行うことができる。すなわち、ノーマルステートにおいてCPU11に供給されるSTPCLK#信号がイネーブルつまりアクティブステートに設定されると、CPU11は、現在実行中の命令が完了後、次の命令を実行すること無く、内部のパイプラインをすべて空にしてから、ストップグラントサイクルを実行して、ノーマルステートからストップグラントステートに移行する。一方、ストップグラントステートにおいてSTPCLK#信号がディスエーブルつまりインアクティブステートに設定されると、CPU11は、ストップグラントステートからノーマルステートに移行し、次の命令の実行を再開する。

【0028】また、ストップグラントステートからストップクロックステートへの移行は、外部クロックCLKを停止することによって瞬時に行われる。ストップクロックステートにおいてCPU11への外部クロックCLKの供給が再開されると、1ms後にCPU11はストップグラントステートに移行する。このようにストップクロックステートからの復帰には時間がかかる問題がある。

【0029】以上のように、ストップグラントステートは、ノーマルステートに比べ非常にローパワーであり、且つSTPCLK#信号によってノーマルステート、つまり命令実行状態に高速に復帰できるという特徴を持っている。

【0030】このため、本システムでは、CPU動作速度を多段階で制御するCPUスロットリング制御機能を、STPCLK#信号によって、ストップグラントステートとノーマルステートとを周期的に切り替えることによって実現している。この場合、ストップグラントステートとノーマルステートとのデューティ比によって、CPU性能が決定される。

【0031】また、CPU11とホスト-PCIブリッジ12の機能により、CPU11はACPI仕様で定義された前述の4つのCPUパワーステートC0~C3を有している。

【0032】パワーステートC0は、命令実行のための通常の動作状態 (ウェイク状態) として使用されるステートである。CPU動作速度を多段階で制御するCPUスロットリング制御機能は、このパワーステートC0内で実行される。

【0033】パワーステートC1~C3は、アイドル時にCPU11をスリープ状態に設定するために利用されるステートであり、C0ステートへの復帰までのレイテンシおよびスリープの深さが互いに異なっている。C1、C2、C3の順で消費電力は小さくなり、C0ステートへの復帰までのレイテンシはC1、C2、C3の順で大きくなる。C1、C2、C3では、命令は実行されない。

【0034】C2ではバススヌープ動作によってキャッシュの整合性が維持されるが、C3ではスヌープ動作は何ら実行されず、キャッシュの整合性は維持されない。ACPI仕様で規定されているCPUパワーステートC0~C3とグローバルシステムステートG0~G2との関係は図5の通りである。

【0035】G0はシステムが動作している状態 (つまりシステムの電源が入っており、ソフトウェアが実行中の状態) であり、このG0ステートの中でCPU11のパワーステートC0~C3は動的に変化される。

【0036】本実施形態では、OSによってシステムアイドルが検出された時にCPU11を移行すべきCPUSリープステートとして、C2およびC3が選択的に利用される。

【0037】C0からC2への移行には、ホスト-PCIブリッジ12のP\_LVL2レジスタ122が用いられる。つまり、CPU11によるP\_LVL2レジスタ122のリードによって、C0からC2へのパワーステートの切替が引き起こされる。C2内で、ハードウェア割り込み (IRQ) やCPUリセットなどのウェイクイベントが発生すると、CPUパワーステートは自動的にC2からC0に復帰する。

【0038】C0からC3への移行には、ホスト-PCIブリッジ12のP\_LVL3レジスタ123が用いられる。つまり、CPU11によるP\_LVL3レジスタ123のリードによって、C0からC3へのパワーステートの切替が引き起こされる。また、C0からC3への移行の際には、ARB\_DISレジスタ124に“1”がセットされ、PCIアービタ121によるPCIバス2のバスアービトレーション動作がディスエーブルされる。これにより、あらたなバスアクセス要求は許可されない。C3内で、ハードウェア割り込みやCPUリセットなどのウェイクイベントが発生すると、CPUパワーステートは自動的にC3からC0に復帰する。

【0039】G1はシステム全体のスリープステートであり、G1ステートには、消費電力の異なるシステムステートS1~Sが定義されている。本システムでは、G1は主にサスペンドやハイバネーション時のシステムステートとして利用されている。

【0040】G2はオフ状態、つまり全てのソフトウェアの実行は終了し、システムの電源が切られている状態である。さらに、図1のCPU11は、次のようなシステム管理機能を備えている。

【0041】すなわち、CPU11は、アプリケーションプログラムやOSなどのプログラムを実行するためのリアルモード、プロテクトモード、仮想86モードの他、システム管理モード (SMM; System Management mode) と称されるシステム管理または電力管理専用のシステム管理プログラムを実行するための動作モードを有している。



【0042】リアルモードは、最大で1Mバイトのメモリ空間をアクセスできるモードであり、セグメントレジスタで表されるベースアドレスからのオフセット値で物理アドレスが決定される。プロテクトモードは1タスク当たり最大4Gバイトのメモリ空間をアクセスできるモードであり、ディスクブリタテーブルと称されるアドレスマッピングテーブルを用いてリニアアドレスが決定される。このリニアアドレスは、ページングによって最終的に物理アドレスに変換される。仮想86モードは、リアルモードで動作するように構成されたプログラムをプロテクトモードで動作させるためのモードであり、リアルモードのプログラムはプロテクトモードにおける1つのタスクとして扱われる。

【0043】システム管理モード（SMM）は疑似リアルモードであり、このモードでは、ディスクブリタテーブルは参照されず、ページングも実行されない。システム管理割込み（SMI ; System Management Interrupt）がCPU11に発行された時、CPU11の動作モードは、リアルモード、プロテクトモード、または仮想86モードから、SMMにスイッチされる。SMMでは、システム管理またはパワーセーブ制御専用のシステム管理プログラムが実行される。

【0044】SMIはマスク不能割込みNMIの一種であるが、通常のNMIやマスク可能割込みINTRよりも優先度の高い、最優先度の割り込みである。このSMIを発行することによって、システム管理プログラムとして用意された種々のSMIサービスルーチンを、実行中のアプリケーションプログラムやOS環境に依存せずに起動することができる。このコンピュータシステムにおいては、前述のCPUパワーステートの切り替えなどのためにこのSMIを利用している。

【0045】主メモリ13は、オペレーティングシステム、処理対象のアプリケーションプログラム、およびアプリケーションプログラムによって作成されたユーザデータ等を格納する。CPU11がSMMに移行する時には、CPUステータス、つまりSMIが発生された時のCPU11のレジスタ等が、主メモリ13の所定のアドレス空間にマッピングされたSMRAMにスタック形式でセーブされる。このSMRAMには、BIOS-ROM17のシステム管理プログラムを呼び出すための命令が格納されている。この命令は、CPU11がSMMに入った時に最初に実行される命令であり、この命令実行によってシステム管理プログラムに制御が移る。

【0046】表示コントローラ14は、画像メモリ（VRAM）141に描画された表示データを本PC本体に設けられたLCD142および外部CRT143の一方に、あるいは双方に表示する。この表示コントローラ14はPCIバス2のバスマスタとして動作することができる。LCD142の輝度は、LCD142のバックライトの輝度を制御するLCD輝度制御ロジック19

によって制御される。

【0047】PCI-ISAブリッジ15は、PCIバス2とISAバス3とをつなぐブリッジであり、PCIバス2のバスマスタとして動作することができる。このPCI-ISAブリッジ15には、SMI発生回路151、CPUスロットリング制御回路152、DMAコントローラ（DMAC）153などが設けられている。

【0048】SMI発生回路151はCPU11にSMI信号を発生する。SMI信号の発生要因には、ソフトウェアSMI、I/OトラップSMI、電源スイッチ操作などがある。ソフトウェアSMIは、ソフトウェアによってアクセス可能なダウンカウンタなどを利用して発生される。つまり、ソフトウェアがSMI信号発生までの時間に相当する値をSMI発生回路151内のダウンカウンタにセットすると、タイムアウト時にSMI信号が発生される。I/OトラップSMIは、予め決められたI/Oアドレスを用いてINまたはOUT命令を実行することによって引き起こされる。

【0049】CPUスロットリング制御回路152は、ストッククロック信号STPCLK#を用いて前述のCPUスロットリング制御を実行するためのものであり、PCI-ISAブリッジ15内のレジスタにセットされた制御情報に基づいて、CPU11の性能をその最大性能値に対して所定の割合に設定する。このCPUスロットリング制御回路152の具体的な構成は図2を参照して後述する。

【0050】DMAコントローラ（DMAC）153は、バスマスタ機能を持たないデバイスと主メモリ13間のDMA転送を実行するものであり、複数のDMAチャネルを有している。

【0051】I/Oコントローラ16は、HDD162などのIDEデバイスを制御するためのバスマスタIDEコントローラ161を内蔵している。バスマスタIDEコントローラ161は、HDD162と主メモリ13との間のデータ転送のためにバスマスタとして動作することができる。また、I/Oコントローラ16は、PC本体に設けられたUSBポートに接続可能な各種USBデバイスを制御する機能も有している。

【0052】BIOS-ROM17は、システムBIOS（Basic I/O System）を記憶するためのものであり、プログラム書き替えが可能なようにフラッシュメモリによって構成されている。システムBIOSは、リアルモードで動作するように構成されている。このシステムBIOSには、システムブート時に実行されるIRTLルーチンと、各種I/Oデバイスを制御するためのデバイスドライバと、システム管理プログラムが含まれている。システム管理プログラムは、SMMにおいて実行されるプログラム（SM-BIOS）であり、CPUスロットリング制御、CPUパワーステートの切り替え制御、LCD輝度制御などを行う。

【0053】電源コントローラ18は、本PCのパワーオン／オフを制御するためのものであり、電源スイッチ183のオン／オフ、バッテリー182の残存容量、ACアダプタ181の接続の有無、ディスプレイパネル開閉検出スイッチのオン／オフなどの状態監視機能を有している。

【0054】図2には、CPUスロットリング制御回路152の具体的な構成の一例が示されている。CPUスロットリング制御回路152には、STPCLK#の発生制御を行うストップクロック制御回路21、STPCLK#の発生間隔を制御するストップクロックインターバルタイマ22、CPU11をストップグラントステートに保持する期間を制御するストップクロックホールドタイマ23、およびCPU11によってプログラム可能なレジスタ群24が設けられている。レジスタ群24には、CPUスロットリング制御の有効／無効を設定するための制御フラグ、およびCPUスロットリングレベルを指定するデューティ制御情報が設定される。

【0055】以下、ストップグラントステートとノーマルステートとをある時間間隔で交互に繰り返すCPUスロットリング制御動作について、図3を参照して説明する。ストップクロックインターバルタイマ22は、レジスタ群126にストップクロックインターバル時間（デューティ幅）がセットされると、その時間毎に定期的にタイムアウト出力を発生する。このタイムアウト出力に応答して、ストップクロック制御回路21はSTPCLK#をアクティブステートに設定する。また、ストップクロックホールドタイマ23は、STPCLK#がアクティブステートに設定されてから、レジスタ群126にセットされたホールド時間（クロックオフタイム）だけ経過した時にタイムアウト出力を発生する。このタイムアウト出力に応答して、ストップクロック制御回路21はSTPCLK#をインアクティブステートに戻す。

【0056】STPCLK#がアクティブステートに設定されると、CPU11はグラントサイクルを実行した後、ノーマルステートからストップグラントステートに移行する。そして、STPCLK#がインアクティブステートに戻されるまで、ストップグラントステートに維持される。

【0057】従って、CPU11はある時間間隔でストップグラントステートとノーマルステートを交互に繰り返すので、その平均動作速度は、スロットリング制御を行わない最大速度の時よりも低下される。この場合、動作速度の低下の割合は、デューティ制御情報（デューティ幅、クロックオフタイム）によって与えられるストップクロックインターバル時間とホールド時間との比によって決定される。したがって、デューティ制御情報によって、STPCLKのデューティ比を図4の様に可変設定することにより、CPUの性能（稼働率）を多段階に可変設定することが可能となる。

【0058】次に、本実施形態で用いられるパワーセーブ制御方法の原理について説明する。本実施形態では以下のパワーセーブ機能が提供される。

#### (1) パワーセーブ1

システムが一定時間以上一定以下の処理しか行っていない場合（システム稼働率が低い状態が続いているとき）、CPU11をスリープ状態からウェイク状態に復帰させるときに、段階的にCPUスロットリングレベルを低い方から高い方に一定間隔で上げる（緩慢復帰）ことにより、パワーセーブを行う。

【0059】ここで、「システム稼働率が低い状態が続いている」ことの判断方法を、図6を参照して説明する。

（スリープ時間  $t_1$  の測定）CPUパワーステートは、OSからのアイドル要求が発行された時にウェイク状態（C0）からスリープ状態（C2またはC3）に移行され、そして、割り込みなどのウェイクイベントが発生した時にウェイク状態（C0）に復帰する。

【0060】スリープ状態に入る前にタイマを読み、スリープ状態から抜け出したらまたタイマを読む。そして、差分からスリープ時間  $t_1$  を得る。ここで、タイマとしては、システムがG0ステートに維持されている期間中カウント動作を続けるタイマ、たとえばACPI仕様で規定されているPMタイマなどを使用すればよい。このタイマはホスト-PCIブリッジ12またはPCI-ISAブリッジ15に設けられている。

【0061】（ウェイク時間  $t_2$  の測定）スリープ状態から抜け出したらタイマを読み、再びスリープ状態に入る前にタイマを読む。差分からウェイク時間  $t_2$  を得る。

【0062】（低システム稼働率の継続時間  $t_3$  の測定） $t_3$  は、 $t_1/t_2$  の比率がある一定値（例えば、 $1/2.6$ ）以下となる状態が継続している時間である。 $t_3$  のカウントアップは、比率が $1/2.6$ 以下が続いている  $t_1$  と  $t_2$  の値を累積していくことによって行われる。

【0063】 $t_3$  が予め決められた時間（ $t_5$ ）を越えると、「システム稼働率が低い状態が続いている」と判断される。つまり、 $t_3$  が  $t_5$  を越えたとき、緩慢復帰処理実行の条件を満たしたとして次のウェイク時から緩慢復帰処理を行う。

【0064】（緩慢復帰）CPU11がスリープ状態に維持されている状態でハードウェア割り込み（IRQ）などのウェイクアップイベントが発生したときは、通常は、図7（A）に示すように、CPU11の動作速度（CPU稼働率）はすぐにスリープ状態移行前の動作速度（CPU稼働率）に復帰されるが、緩慢復帰の条件が満足されているときは、図7（B）に示すように緩慢復帰処理が実行され、前述のCPUスロットリング制御によって所定の時間間隔で段階的にCPU動作速度が低レ

ベルから所定の高レベルにまで徐々に上昇される。

【0065】これにより、CPU11の動作速度をすぐにスリープ状態移行前の動作速度に復帰させる場合に比し、図7(B)に斜線で示されている分だけ、CPU11の電力消費を少なくすることが可能となる。

【0066】なお、緩慢復帰処理で最終的に戻されるCPU性能はスリープ状態移行前の性能までであり、図8に示すように、バッテリーの残存容量やCPU温度などの条件によって例えばCPU稼働率=50%の性能で動作している状態からスリープ状態に移行した場合には、緩慢復帰処理で最終的に戻されるのはCPU稼働率=50%の性能までとなる。

【0067】また、このような緩慢復帰処理を行った場合でも、C0へのウェイクタイミング自体が遅れることはない。したがって、スリープ状態の期間中にたとえばユーザによってキーボードやマウスの操作が行われた場合でも、即座にそれに対応する処理を開始することができる。また、ある一定期間後にはCPU性能は、スリープ状態に入る前に設定されていた性能に復帰される。このため、過大なCPU負荷を伴う処理が突然要求されない限りは、ユーザによる体感速度が低下されるといった不具合は一切生じない。よって、システム性能の低下を最小限に抑えつつ十分なパワーセーブを実現することができ、バッテリーによる動作時間を大幅に延長することが可能となる。

【0068】なお、一旦、緩慢復帰条件が満たされた後もt1とt2の測定は続けられる。t1とt2の比率が、1/26よりも増えた場合には、t3をクリアし、以後、再び累積されたt3の値がt5を越えるまで、緩慢復帰処理は実行されない。

#### 【0069】(2) パワーセーブ2

システムが特定の処理(DMACやバスマスタIDEによるバスマスタ動作)を行っていない場合には、アイドル時に入るCPUステートを、C2ではなくC3に持っていくことにより、パワーセーブを行う。

【0070】すなわち、アイドル時には、そのときのシステム状態がC3への移行条件を満足しているか否かが判断される。移行条件を満足していれば、より深いスリープステートC3(ディープスリープ)に移行し、移行条件を満足していなければ、C0への高速復帰が可能なスリープステートC2(クイックスタート)に移行する。

【0071】C3に移行するための条件は、CPU11以外の他のデバイスがバスマスタ動作していないことである。前述したようにC3ではCPUキャッシュと主メモリ13との間の整合性を維持するためのスヌープ動作が行われなため、もしバスマスタ動作しているデバイスが存在すると、そのバスマスタ動作しているデバイスによる主メモリ13の書き換えによって主メモリ13とCPUキャッシュの不整合が生じる危険があるためであ

る。

#### 【0072】(3) パワーセーブ3

バッテリー残量がある一定基準以下になったら、CPUスロットリングをより深く効かせる(スロットリングレベルを下げる)ことにより、ウェイク状態におけるCPU性能を下げたパワーセーブを行う。

#### 【0073】(4) パワーセーブ4

バッテリー残量がある一定基準以下になったら、LCDの輝度をより低くすることにより、パワーセーブを行う。

【0074】図9には、OSによってシステムアイドルが検出されてから、SM-BIOSによってCPUスリープ制御が実行されるまでの流れが示されている。OSは、CPU11が実行すべきタスクが無くなると、システムアイドル状態(CPUアイドル)であることを検出し、システムBIOS(またはSM-BIOS)に対してアイドル要求を発行する。このアイドル要求を受けて、システムBIOS(またはSM-BIOS)は、CPU11をウェイク状態からスリープ状態に移行させるためのCPUスリープ制御を開始する。

【0075】次に、図10のフローチャートを参照して、BIOSによる処理手順を説明する。システムBIOSは、まず、PCがバッテリー駆動中であるか否かを判断する(ステップS11)。これは、ACアダプタが接続されているか否かを示す情報を電源コントローラ18から取得することによって判断される。ACアダプタが接続されてなければ、バッテリー駆動中である。

【0076】バッテリー駆動中の場合には、以下のステップS12~S21の処理に移行する。すなわち、まず、これまでに累積されている比率1/26以下の“t1”、“t2”の積み上げ値により時間“t3”を取得し(ステップS12)、その“t3”がしきい値“t5”を越えているか否かが判断される(ステップS12)。“t3”がしきい値“t5”を越えていれば、システム稼働率が低い状態が続いていると認識される。この場合には、まず、タイマを読み、前回のウェイク時に読んだタイマ値との差分から、“t2”を計測する(ステップS14)。次いで、次のウェイク時に緩慢復帰処理を実行するために、CPUスロットリング制御をイネーブルにすると共に、デューティ制御情報によってスロットリングレベルを最低レベル(例えば12.5%)に設定する(ステップS15)。そして、CPUパワーステートをC0から、C2またはC3に切り替えるためのCPUスリープ処理を実行する(ステップS16)。BIOSの処理はこの時点で中断される。

【0077】CPUパワーステートがC2またはC3の状態、PC内のデバイスからのハードウェア割り込み(Irq)などのウェイクイベントが発生すると、CPUパワーステートは自動的にC0に復帰する。この場合、CPU性能はステップS15で設定した最低レベル(12.5%)に復帰される。そして、C2またはC3

移行前に実行した命令の次の命令から、CPU11は命令の実行を開始する。これにより、BIOSのステップS17から処理が再開される。BIOSは、タイマを読み、ステップS14で読んだタイマ値との差分から“t1”を計測する(ステップS17)。次いで、定期的にCPUスロットリングレベルを更新するために、次回BIOSを呼び出すまでの時間(t4)を示すタイマ値をソフトウェアSMIタイマに設定した後(ステップS18)、復帰(リターン)命令を実行して、OSに制御を戻す。

【0078】CPU11がウェイク状態(パワーステートC0)の期間に、デューティ情報の変更を要求するソフトウェアSMIが発生すると、SM-BIOSの緩慢復帰処理ルーチンが起動される。緩慢復帰処理ルーチンは、図11に示すように、デューティ制御情報を更新することによりCPUスロットリングレベルを+1レベル上昇させる(ステップS31)。+1レベルアップしたCPUスロットリングレベルが最終的に戻すべきCPUスロットリングレベルに到達していない場合には(ステップS32のNO)、再びソフトウェアSMIの

設定を行う(ステップS33)。これにより、CPUスロットリングレベルが、最終的に戻すべきCPUスロットリングレベルに到達するまで、ソフトウェアSMIによって緩慢復帰処理ルーチンが定期的に呼び出されることになる。

【0079】図10に戻り、ステップS13で、“t3”がしきい値“t5”以下であった場合、つまり緩慢復帰実行の条件が満たされない場合には、ステップS19、S20、S21が実行される。ステップS19、S20、S21は前述のステップS14、S16、S17にそれぞれ対応している。すなわち、前述のステップS15、およびステップS18がスキップされることになる。この場合には、通常復帰処理となる。

【0080】また、ステップS11でAC駆動中であることが検出された場合には(ステップS11のNO)、CPUスリープ処理(ステップS22)のみが実行される。なお、AC駆動中の場合でも、“t1”、“t2”の計測を行うようにしても良い。

【0081】次に、図12のフローチャートを参照して、ステップS16、S20、S22で実行されるCPUスリープ処理の手順を説明する。CPUスリープ処理では、まず、現在のシステムの状態、つまり、DMAチャネルの設定されているか否か、バスマスタIDEがアクティブであるか否か、バスマスタリクエストが発生しているか否か、などをその設定のために使用されるレジスタの値を参照してそれぞれ確認し(ステップS41)、CPU11以外の他のデバイスがバスマスタ動作しているか否かによって、C3への移行条件を満たすか否かを判断する(ステップS42)。なお、この条件に、バッテリー駆動されていること、USBデバイスが接

続されていないこと、等を加えても良い。この場合、移行条件を満たすか否かの判断処理は、図13のステップS51～S55で行われることになる。

【0082】図12のステップS4にて、C3への移行条件を満たしていると判断されると、CPUパワーステートはC0からC3に切り替えられる(ステップS43)。この場合、前述したようにバスアービトレーションも禁止される。一方、C3への移行条件が満足していない場合には、CPUパワーステートはC0からC2に切り替えられる(ステップS44)。

【0083】このように、BIOSの制御の下で、システム状態に応じてスリープステートを動的に切り替えることにより、ACPI-OSのない環境下でも最適なパワーセーブを実現できる。

【0084】なお、バスマスタ動作の有無は、特定のドライバが動作しているか否かを調べることによってもチェックすることができる。また、システム稼働率が低い状態が続いていることを、C3への移行条件に加え、システム稼働率が低い状態が続いている場合にのみ、C3へ移行するための他の条件を満たすか否かを判断しても良い。

【0085】さらに、システム性能を段階的に低レベルから高レベルに上昇させるという本発明の緩慢復帰処理は、システム性能を即座に基の状態に戻すのではなく、段階的に戻すことでパワーセーブを図ることが肝要であるので、CPU速度を段階的に上昇させる制御のみならず、例えば、CPU以外の他の各デバイスのパワーステート制御やクロックの断続的な供給制御を行うことによってそれらデバイスの平均動作速度を段階的に上昇させる制御を行ったり、あるいはキャッシュの有効/無効の断続的な制御や、HDDのモータオン/オフの断続的な制御などを行っても良い。

【0086】また、OSからアイドル要求を受けた場合のみならず、キー入力が一定期間以上無い場合にシステムアイドルと判断して前述のCPUスリープ制御を行うようにしても良い。

【0087】

【発明の効果】以上説明したように、本発明によれば、システム性能の低下を最小限に抑えつつ、十分なパワーセーブを実現できるようになり、バッテリーによる動作時間を大幅に延長することが可能となる。

【図面の簡単な説明】

【図1】本発明の一実施形態に係るコンピュータシステムの構成を示すブロック図。

【図2】同実施形態のコンピュータシステムに設けられたCPUスロットリング制御回路の構成を示すブロック図。

【図3】同実施形態で用いられるCPUスロットリング制御動作を説明するためのタイミングチャート。

【図4】同実施形態のCPUスロットリング制御動作で

【図12】同実施形態においてBIOSによって実行されるCPUスリープ処理の手順を示すフローチャート。

【図13】図12のCPUスリープ処理内で実行されるC3状態への移行条件判別処理の一例を示すフローチャート。

【符号の説明】

1 1 CPU

1 2...ホスト-PCIブリッジ

13…主メモリ

## 10 1.5…PCI-ISAブリッジ

17...BIOS-ROM

1 2 1...PCIアービタ

## 1.5.1...S M I 発生回路

152...CPUスロットリング制御回路

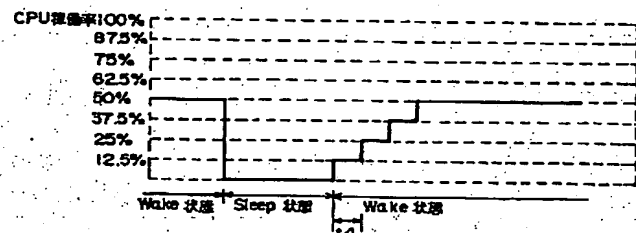
1 5 3 ...DMAコントローラ

## 2 1…ストップクロック制御回路

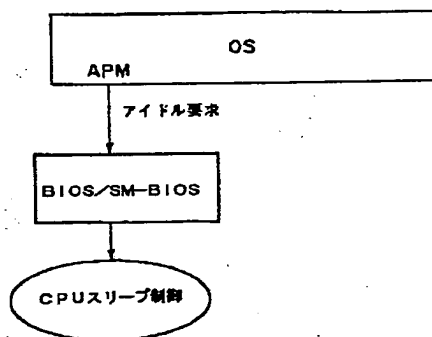
## 22…ストップクロックインターバルタイマ

## 23…ストップクロックホールドタイマ

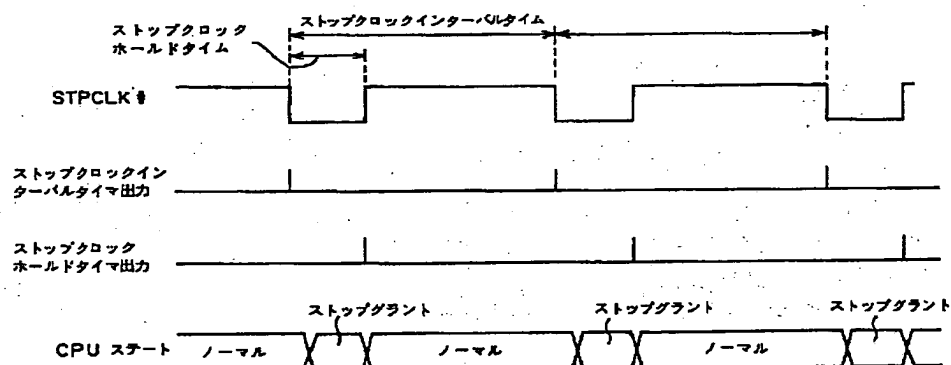
【图 6】



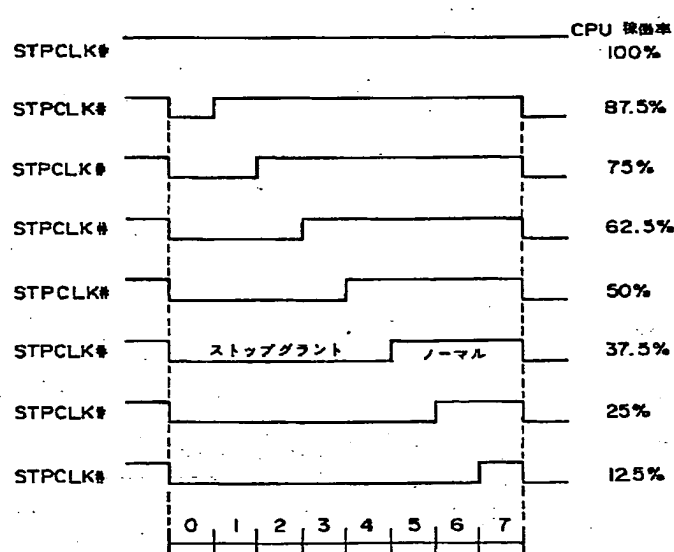
【图9】



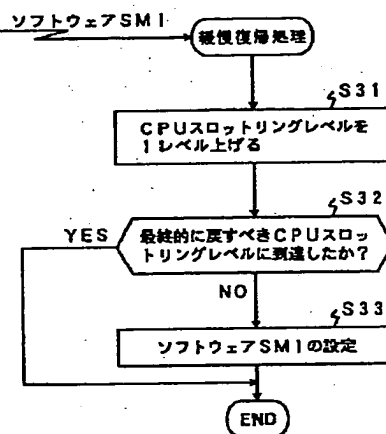
【図 3】



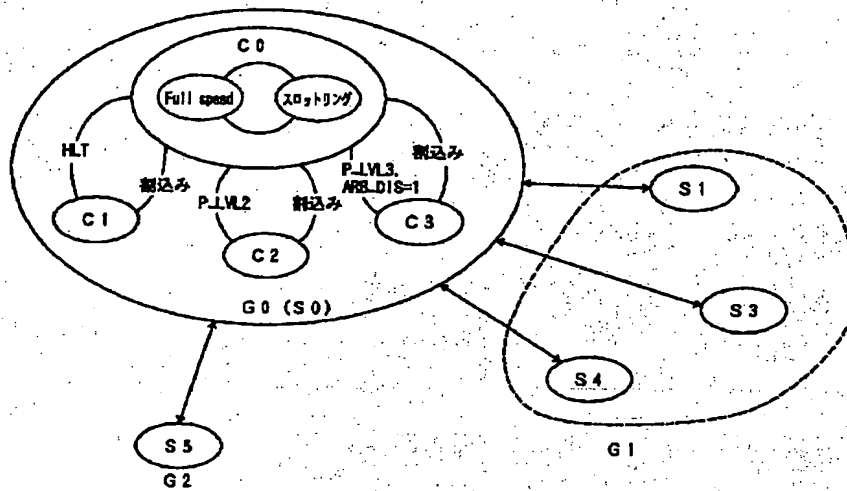
【図4】



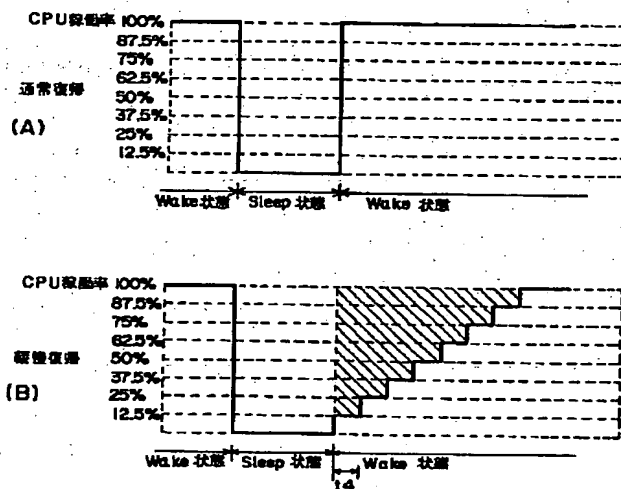
【図 1.1】



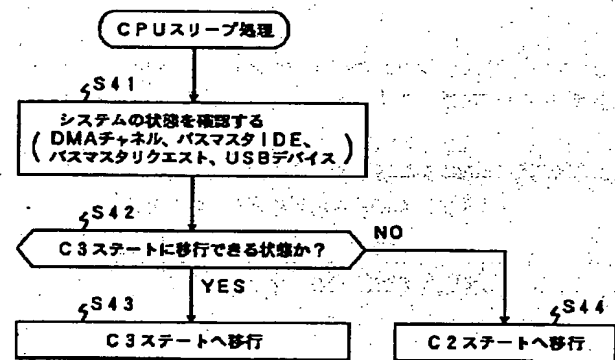
【図5】



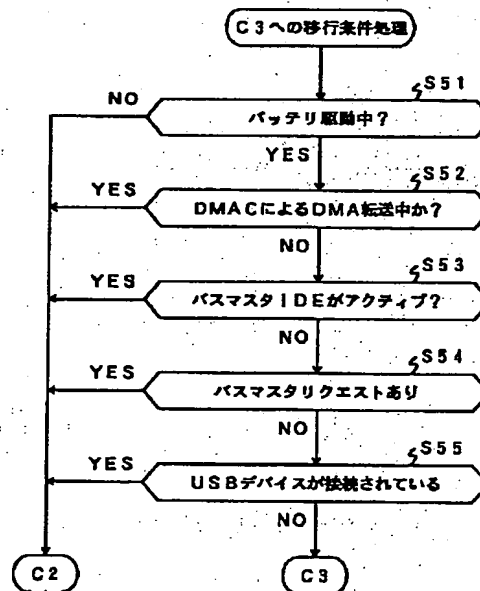
【図7】



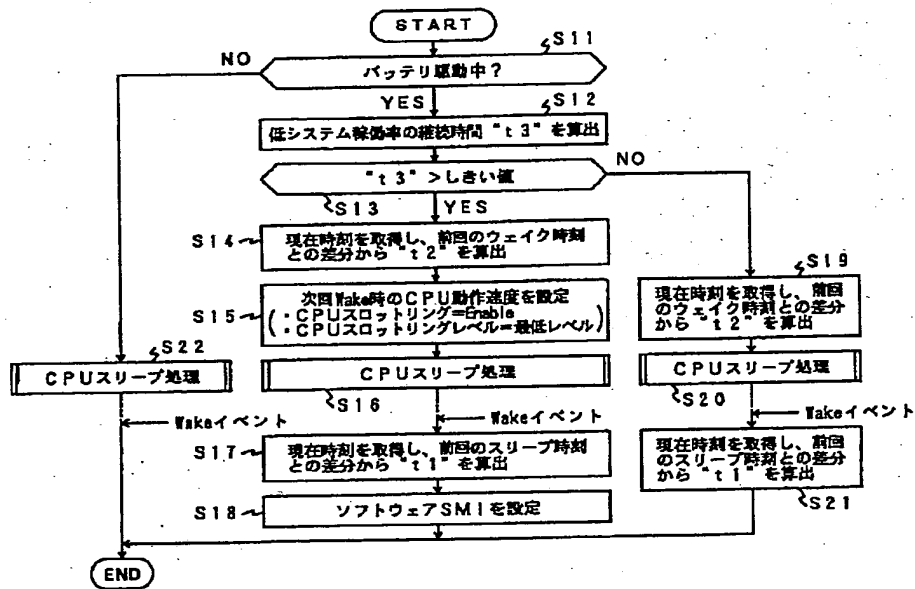
【図12】



【図13】



【図 10】



フロントページの続き

Fターム(参考) 5B011 DA02 EA04 LL08 LL10 LL11  
 5B062 AA05 DD05 HH02 HH06 HH07  
 JJ10  
 5B079 BA01 BB02 BC01 DD02 DD20